

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**UNA HERRAMIENTA PARA VISUALIZAR LA
EVOLUCIÓN DE LOS ESTUDIANTES EN MOOCs:
FACILITANDO SU SEGUIMIENTO**

Fernando Yerón Izquierdo
Tutora: Rosa María Carro Salas

Junio 2018

Una herramienta para visualizar la evolución de los estudiantes en MOOCs: facilitando su seguimiento

AUTOR: Fernando Yerón Izquierdo

TUTORA: Rosa María Carro Salas

**Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2018**

Resumen

Los cursos MOOCs (Massive Online Open Courses) están revolucionando el aprendizaje. Estos cursos permiten el estudio a distancia online para personas que no pueden permitirse una instrucción presencial. De esta forma, cada alumno puede aprender interactuando con los contenidos del curso a su propio ritmo, pudiendo priorizar las razones por las que realiza dicho curso. También estos cursos contienen un gran número de estudiantes. Por ello, se hace muy complicado para los docentes realizar un seguimiento detallado de los alumnos.

Por cada interacción que realiza el alumno en estos cursos, se genera un registro de ella, siendo posible su consulta. Dada la cantidad de datos que se generan, no es sencillo para los docentes visualizarlos ni analizarlos, a menos que usen herramientas específicamente desarrolladas para tal fin. Gracias a la gran cantidad de datos registrados, sí es posible su procesamiento y monitorización automáticas, su análisis e incluso su utilización para predecir situaciones de riesgo de fracaso o abandono de los estudiantes durante la realización del curso.

Este trabajo se centra en facilitar la visualización y el análisis de los datos generados a partir de las interacciones de los estudiantes con los cursos MOOCs. El objetivo es dar soporte al seguimiento de los estudiantes de modo que los docentes puedan localizar situaciones de riesgo a través de una aplicación web, en la que se mostrará información sobre todos los estudiantes y sus trayectorias durante la realización del curso, mediante distintos tipos de tablas y gráficas. De esta manera, los profesores contarán con información sobre la evolución y el rendimiento del alumnado, y podrán llevar a cabo las acciones que consideren oportunas a la vista de esta información.

Palabras clave

Visualización de datos, interfaz gráfica, seguimiento de estudiantes, MOOCs, modelo vista-controlador, procesamiento de datos.

Abstract

MOOCs courses (Massive Online Open Courses) are revolutionizing learning and education. These courses allow online- distance study for people who cannot afford a common or a face-to-face education. In this way, students can learn by interacting with the contents of the course at their own pace, being able to prioritize the reasons why the course was conducted. These courses also contain a huge number of students. Therefore, it is very complicated for teachers to follow up the students in detail.

For each interaction that the students perform in these courses, a record is generated and it is possible to enquire about it. Due to the amount of the data that is generated, it is not easy for teachers to visualize or analyze them, unless they use tools specifically developed for that purpose. Thanks to the large amount of recorded data, it is possible to process and monitor it automatically, analyze it and even use it to predict situations of risk of failure or students drop out during the course.

This work focuses on facilitating the visualization and analysis of the data generated from the interactions of students with MOOCs courses. The objective is to support the students monitoring and track record so that teachers can identify risk situations through a web application, which will show information about all students and their trajectories during the course, through different types of tables and graphs. In this way, teachers will have information on the evolution and performance of the students, and they will be able to carry out the actions that they consider appropriate in view of this information.

Keywords

Data visualization, graphical interface, student monitorization, MOOCs, view-controller model, data processing.

“Do it. Or do not. There is no try” Yoda

Agradecimientos

En primer lugar, agradezco a Rosa Carro por darme la oportunidad de desarrollar esta idea, por su tiempo y su tutela.

A mis padres por ser mi motivación y el espejo en el que fijarme día a día. Por darme fuerzas y ánimos en los peores momentos, cuando creía que no podría con ello.

Finalmente, a mis compañeros Fran y Diego por su ayuda y en general a todos mis amigos que han estado continuamente dando ánimo y haciendo que esto sea posible.

Muchas gracias a todos.

Fernando.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	2
2	Estado del arte	3
3	Análisis y Diseño.....	5
3.1	Análisis de requisitos.....	5
3.2	Archivos de datos	6
3.3	Patrón MVC.....	6
3.4	Visualización de la información	7
3.5	Propuesta de gráficos.....	9
3.6	Diseño de base de datos.....	10
4	Desarrollo	13
4.1	Tecnologías utilizadas	13
4.2	Programación.....	14
4.2.1	Vistas	14
4.2.2	Base de datos de prueba	15
4.2.3	Páginas dinámicas	15
4.2.4	Datos reales	19
4.2.5	Rutas y accesos.....	19
5	Integración, pruebas y resultados	21
5.1	Integración.....	21
5.2	Pruebas	21
5.3	Resultados.....	22
6	Conclusiones y trabajo futuro.....	29
6.1	Conclusiones.....	29
6.2	Trabajo futuro	29
	Referencias	31
	Glosario	33
	Anexos.....	I
A	Capturas de la aplicación.....	I

INDICE DE FIGURAS

FIGURA 1: ESQUEMA MVC [7]	7
FIGURA 2: DIAGRAMA DE CASOS DE USO	9
FIGURA 3: DIAGRAMA DE BASE DE DATOS	10
FIGURA 4: BOCETO DE UNA PANTALLA	15
FIGURA 5: MODELO ALUMNO	16
FIGURA 6: CONTROLADOR ALUMNOS	17
FIGURA 7: PLANTILLA PRINCIPAL	17
FIGURA 8: CONTENIDO DE LA VISTA DE ALUMNOS	18
FIGURA 9: DIAGRAMA DE SECUENCIA DE LA VISTA APROBADOS	19
FIGURA 10: RUTAS DE ALUMNOS Y SEGUIMIENTO DETALLADO	19
FIGURA 11: EJEMPLO DE PRUEBA UNITARIA	22
FIGURA 12: EJEMPLO DE PRUEBA DE APLICACIÓN	22
FIGURA 13: PANTALLA DE INICIO DE SESIÓN	23
FIGURA 14: PANTALLA ELECCIÓN DE CURSO	23
FIGURA 15: PANTALLA PRINCIPAL	24
FIGURA 16: PANTALLA DE SEGUIMIENTO.....	24
FIGURA 17: PANTALLA DE DATOS DE ALUMNOS.....	25
FIGURA 18: PANTALLA DE PORCENTAJE DE APROBADOS.....	25
FIGURA 19: EJERCICIOS GENERAL	26
FIGURA 20: EJERCICIOS INDIVIDUAL.....	26
FIGURA 21: ACCESOS GENERAL	27
FIGURA 22: COMPARATIVA DE ACCESOS	27
FIGURA 23: PANTALLA PRINCIPAL	I
FIGURA 24: RESETEO DE CONTRASEÑA.....	I

FIGURA 25: REGISTRO SOLO PARA ADMINISTRADORES	II
FIGURA 26: SEGUIMIENTO INDIVIDUAL.....	II
FIGURA 27: COMENTARIOS GENERAL	III
FIGURA 28: COMENTARIOS INDIVIDUAL	III
FIGURA 29: PERFILES GENERAL 1	IV
FIGURA 30: PERFILES GENERAL 2	IV
FIGURA 31: PERFILES GENERAL 3	V
FIGURA 32: PERFILES INDIVIDUAL	V
FIGURA 33: ACCESOS INDIVIDUAL	V
FIGURA 34: ACTIVIDADES GENERAL 2	VI
FIGURA 35: ACTIVIDADES GENERAL 3	VI
FIGURA 36: ACTIVIDADES INDIVIDUALES 2	VI
FIGURA 37: VIDEOS GENERAL.....	VII
FIGURA 38: VIDEOS INDIVIDUAL.....	VII
FIGURA 39: ACTIVIDAD GENERAL.....	VIII
FIGURA 40: ACTIVIDAD INDIVIDUAL.....	VIII

1 Introducción

1.1 Motivación

A finales de los años 80 la enseñanza universitaria online se consideraba más una enseñanza por correspondencia [1]. Esta ideología empezó a cambiar cuando surgió la idea de poner a disposición de la comunidad universitaria de forma gratuita los recursos educativos. El desarrollo de las tecnologías de entonces fueron permitiendo lo que hoy se conocen como Massive Online Open Courses o MOOCs. La tecnología de hoy en día permite el aprendizaje online de alta calidad, esto se debe el acceso a todos los recursos (transparencias, libros, videos, apuntes, ...) se puede realizar online [2].

La gran cantidad de estudiantes y el hecho de que cada uno pueda ir a su propio ritmo dificultan la realización de un seguimiento individualizado de la evolución de cada estudiante por parte del profesor responsable del MOOC. Por ello, parece conveniente la creación de herramientas que faciliten esta labor al profesorado. No obstante, una mera visualización de la evolución de cada estudiante no es suficiente cuando se trata de cursos con cientos o incluso miles de estudiantes. En investigaciones se han desarrollado mecanismos para la predicción de casos de abandono o de fracaso por parte de los alumnos. Este tipo de trabajos surgen motivados por la mencionada dificultad de hacer un seguimiento detallado tanto individual como colectivo de los alumnos por parte del docente [3].

Sería de gran utilidad disponer de una herramienta que permitiera no solo la visualización de información (datos de interacciones de los estudiantes), si no que facilitara la identificación de situaciones de riesgo y ofreciera esa información de forma visual, de manera que el docente pudiera conocer de forma sencilla qué estudiantes están en situación de riesgo para tomar las medidas que considere pertinentes.

De este modo surge la motivación para este trabajo: se trata de diseñar y desarrollar una herramienta que dé soporte a los profesores de este tipo de cursos para que puedan e identificar rápidamente aquellos casos para los que la predicción indica riesgo de fracaso, siendo posible analizar el origen del problema y, si se considera oportuno, tomar las acciones oportunas para remediarlo, si fuera el caso.

1.2 Objetivos

El objetivo principal de este trabajo es la creación de una herramienta fácil de utilizar e intuitiva que permita visualizar las acciones y la evolución de los estudiantes de los MOOCs a lo largo del tiempo, así como las predicciones sobre el riesgo de abandono o de fracaso. Para ello, se han planteado los siguientes objetivos:

- Analizar el problema e identificar los requisitos que debe tener la herramienta. Realizar unos bocetos que reflejen la idea inicial de la aplicación.
- Diseñar la herramienta.
- Programar un primer prototipo de la herramienta.
- Crear una base de datos con datos de prueba.
- Modificar la herramienta para que sea interactiva y utilice la base de datos creada.
- Extraer datos de un MOOC real e introducirlos en la base de datos.

- Realizar pruebas para comprobar el correcto funcionamiento de la herramienta.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1: en él se presenta el contexto del TFG y el problema planteado. Se motiva la necesidad de este trabajo y se describen trabajos relacionados con él.
- Capítulo 2: en este capítulo se describen el análisis realizado. También se muestra como ha sido diseñada la API, es decir, las técnicas utilizadas para resolver el problema planteado.
- Capítulo 3: explica cómo se ha desarrollado la aplicación, de qué fases se ha compuesto el desarrollo de la API y las tecnologías que han sido necesarias.
- Capítulo 4: este capítulo recoge la explicación de cómo se han integrado los datos en la aplicación, las pruebas que se han realizado para comprobar su correcto funcionamiento y el resultado final de la misma.
- Capítulo 5: se exponen las conclusiones a las que se ha llegado después de realizar el TFG y se proponen futuros trabajos.

Finalmente se incluye la bibliografía y un glosario de términos y el anexo I, en el que se muestran las capturas de pantalla de la aplicación.

2 Estado del arte

Este trabajo se enmarca en el ámbito de la visualización y monitorización de la información relacionada con las interacciones de los estudiantes con los MOOCs y con la predicción sobre su desempeño. A continuación, se describen algunos de los trabajos más directamente relacionados con este.

La herramienta Analyse [4] es una herramienta desarrollada en Open edX y utilizada por la Universidad Carlos III de Madrid. Esta herramienta obtiene datos de bajo nivel y los transforma en indicadores que son visualizados.

Los indicadores utilizados son: la distribución del tiempo durante la realización del curso, las evaluaciones realizadas, los vídeos vistos, los accesos al curso y el tiempo que invierten en el mismo.

Algunas características de esta herramienta son:

- Transforma datos de registros en indicadores significativos para el aprendizaje.
- Permite la visualización relacionada con ejercicios, vídeos y la actividad en general del curso.
- Los profesores tienen dos formas de acceso a la información: de forma general o individual. De esta forma el profesor puede hacer una monitorización de cómo progresa el curso en global y también a su vez controlar a cada estudiante.
- Los alumnos de los cursos pueden acceder a su información con el objetivo de motivar su aprendizaje, que reflexionen y hagan autocrítica sobre su desarrollo en el curso.
- Los indicadores son procesados en segundo plano y durante intervalos de tiempo regulares.

Esta herramienta se basa en resultados como son actividades y ejercicios para poder realizar la monitorización. Desde mi punto de vista se considera más completo el uso de aprendizaje automático para detectar situaciones de riesgos durante el aprendizaje del alumno. Además, la visualización de sus variables se centra en gráficos de barras, que dependiendo del docente pueden no resultarles claros. Por ello en este estudio se añaden distintos tipos de gráficos a las variables para ofrecer distintos tipos de vistas que puedan facilitar la comprensión por parte del docente.

Por otro lado, en [5] se presenta una aplicación web que utiliza los registros generados por la plataforma de aprendizaje Selene para realizar un seguimiento de las actividades (Estrategia de seguimiento a las actividades de aprendizaje de los estudiantes en cursos en línea masivos y privados (MPOC) con reconocimiento académico en la Universidad del Cauca). Pese a estar diseñado para MPOCs (cursos privados) en vez de para MOOCs (cursos gratuitos), la idea de fondo es la misma: todas las interacciones de las realizadas por los estudiantes y ofrecer visualizaciones.

Las variables que se visualizan son las siguientes:

- Ingresos en el curso. De esta forma el docente sabrá si el proceso para la inscripción de los estudiantes es el correcto y a su vez si los estudiantes habían recibido las orientaciones pertinentes para el comienzo del curso.
- Actividades en vídeos. La plataforma permite ver las acciones que ha realizado cada alumno con los vídeos, la fecha y hora en que ha sido realizado, todo ello en formato de tabla.

- Actividades en foros. Al igual que con los vídeos, en una tabla se recoge la participación de los estudiantes en los foros.
- Interacción con evaluadores. En todo curso se realizan pruebas o exámenes, sobre todo si la modalidad es no presencial. La aplicación recoge el número de estudiantes que se han presentado a los exámenes, el número de intentos y repeticiones realizadas durante la resolución de cada uno de ellos.
- Informes automáticos. Esta funcionalidad permite obtener la información de la base de datos y volcarla en distintos archivos de texto.

Esta herramienta recoge mayor cantidad de datos de interés que el trabajo anterior. Pese a ser más completo respecto a este aspecto, la forma de visualización es muy basta. Es interesante realizar visualización de datos mediante tablas, pero se hace más tediosa, por eso la utilización de gráficas añadidas a esta tabla podría ser una mejora bastante interesante.

Por último, el trabajo [6] (Estrategia para el seguimiento y evaluación de los aprendizajes en un MOOC de introducción al álgebra) propone una forma de seguimiento por parte del docente en el ámbito del álgebra. Este seguimiento consiste en un *feedback* generado automáticamente que informa si las respuestas a los ejercicios propuestos son válidas y contiene comentarios sobre la posible fuente del error, con la finalidad de ayudar al estudiante con el curso. Estos ejercicios se ofrecen en forma de videos y problemas de autoevaluación que son de alta ayuda para el estudiante, y se apoya en la experiencia de una primera edición del curso, actitudes y expectativas de los alumnos participantes. En este estudio el uso de cuestionarios de autoevaluación se considera positivo, ya que el estudiante puede regular su aprendizaje, detectar sus problemas e intentar resolverlos.

Pese a que este estudio está enfocado solo al curso de Introducción al Álgebra, se considera de interés ya que propone una forma distinta de seguimiento en un MOOC.

3 Análisis y Diseño

3.1 Análisis de requisitos

A continuación, se detallan los requisitos funcionales del sistema que han sido identificados, que deberán ser cumplidos para satisfacer las necesidades planteadas.

- **RF1:** El usuario podrá acceder a la aplicación introduciendo su correo y contraseña.
- **RF2:** No se permitirá el acceso al resto de la aplicación si no se ha iniciado sesión previamente.
- **RF3:** Se permitirá cerrar sesión al usuario desde la barra de navegación, pulsando el botón del nombre.
- **RF4:** El sistema guardará las contraseñas del usuario cifradas.
- **RF5:** El profesor podrá realizar un seguimiento de todos los cursos que imparta.
- **RF6:** Se podrá visualizar de forma rápida la cantidad de aprobados y suspensos de un curso.
- **RF7:** Permitirá visualizar las interacciones que realizan los estudiantes con los videos del curso.
- **RF8:** Forma de visualizar las interacciones que realizan los estudiantes con las actividades propuestas.
- **RF9:** Forma de visualizar el número de comentarios realizados en el foro del curso.
- **RF10:** Forma de visualizar el número de interacciones que ha realizado un alumno en el curso.
- **RF11:** Forma de visualizar el número de accesos que realizan los estudiantes en el curso.
- **RF12:** Forma de visualizar el estado de completo del perfil del estudiante.
- **RF13:** Todas las **formas** de visualización tendrán que ser **individuales** (pertenecientes a un estudiante dado) y **generales** (datos agrupados que representen el estado del curso en su globalidad), exceptuando el porcentaje de aprobados.
- **RF14:** Forma de visualizar todos los datos de un alumno y la predicción de aprobado o suspenso para dicho alumno.
- **RF15:** Forma de comparar el rendimiento de dos alumnos mediante la comparación de cualquiera de las gráficas anteriores.

Por otra parte, se han identificado los siguientes requisitos relacionados con la interfaz y la interacción con el sistema.

- **RNF1:** La primera pantalla que el usuario verá contendrá el título de la aplicación “Seguimiento de MOOCs” con el botón de acceso.
- **RNF2:** Una vez dentro de la aplicación, todas las pantallas tendrán una barra lateral que permitirá la navegación más rápida entre pantallas.
- **RNF3:** Todas las pantallas de la aplicación tendrán una barra de navegación en la parte superior, que incluirá también el nombre de la aplicación.
- **RNF4:** En la parte derecha de la barra de navegación aparecerá el nombre del usuario que ha iniciado sesión.
- **RNF5:** La aplicación estará soportada por los navegadores: Chrome y Firefox.
- **RNF6:** Número de usuarios elevados al mismo tiempo. Pese a que esta aplicación está planteada solo para los profesores, puede recoger una gran cantidad de cursos y sería beneficioso tener concurrencia en la aplicación.

- **RNF7:** En caso de producirse algún fallo se podrá notificar mediante correo electrónico al administrador del sistema.
- **RNF8:** Si se produjesen dudas acerca del uso hay un contacto mediante correo electrónico donde se podrán exponer esas dudas.
- **RNF9:** Debido a la gran cantidad de datos que recogen los MOOCs se necesitarán servidores potentes, e incluso una granja de servidores debido a que la aplicación podrá recoger varios cursos lo que conlleva mucha información.

3.2 Archivos de datos

La información con la que va a trabajar la herramienta para su visualización será aquella disponible en los logs de los MOOCs generados en EdX.

Los datos están organizados en archivos csv. A continuación, se enumeran los archivos y los datos que contiene cada uno de ellos:

- **Profile<id_curso>.csv:** Aquí se encuentra el perfil del alumno de forma anónima. Podemos encontrar variables de interés como:
 - Id del usuario.
 - Idioma.
 - Localización.
 - Género.
 - Fecha de nacimiento.
 - Nivel de educación.
 - Metas.
 - País.
 - Ciudad.
 - Imagen de perfil.
- **Social<id_curso>.csv:** Recoge los comentarios realizados por los alumnos de forma anónima en los foros.
- **Certificates<id_curso>.csv:** Recoge la generación de certificados de los alumnos de forma anónima.
- **Eventos:** Aquí se encuentran los logs en archivos de texto con todas las interacciones que realizan los alumnos con el curso. Estos logs tienen varias variables, de las que destacan para el uso de nuestra aplicación las siguientes:
 - Id del usuario “alumno”.
 - Id del curso.
 - Tipo de evento.
 - Nombre del evento
 - Fecha.

Más adelante, concretamente en el apartado 5.1 se explicará en detalle cómo se tratan estos archivos para extraer los datos a integrar en la base de datos y en la aplicación.

3.3 Patrón MVC

Para la realización de esta aplicación se ha utilizado el patrón conocido como Modelo-Vista-Controlador (MVC).

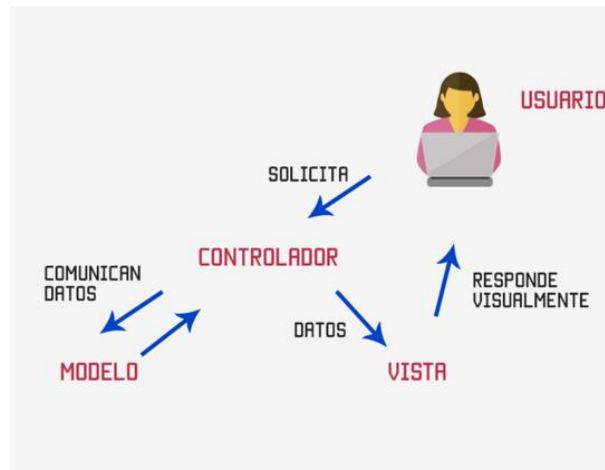


Figura 1: Esquema MVC [7]

El patrón MVC [8] da la posibilidad de separar por módulos el código dependiendo de la función que se realice, de esta forma quedan tres: modelo, vista y controlador.

- **Modelo:** Se encarga de la parte de los datos, es decir, de la comunicación con la base de datos y de las consultas.
- **Controlador:** Es el módulo que se encarga de recibir las peticiones del usuario y pide recursos al modelo para finalmente comunicárselos al módulo vista.
- **Vista:** Es la parte visual, es decir, la interfaz gráfica a través de la cual se muestra la información.

La elección de este patrón de diseño se debe a la facilidad y modularidad que ofrece este sistema. También hace más sencilla la realización de pruebas y la reutilización de módulos. Se podría considerar como inconveniente que al tener modularidad hay se aumenta la cantidad de ficheros [9].

3.4 Visualización de la información

A continuación, se describen las acciones que puede realizar el usuario (profesor) desde la aplicación para el seguimiento del alumno, y el modo en que se visualizará la información en cada caso.

- **Datos alumno:** Podrá visualizar los datos relevantes de todos los alumnos en formato de tabla, pudiendo filtrar por columnas o realizar búsquedas. La columna final de esta tabla consistirá en la predicción de aprobado o suspenso para el alumno. Para una mejor visualización de este dato, se utilizará un código de colores: verde para aprobado y rojo para suspenso.
- **Gráficas individuales:** Podrá acceder a las gráficas de cada alumno a través de la tabla anterior, y seleccionar alumnos para así comparar sus gráficas.
- **Aprobados/suspensos:** Podrá ver un gráfico de tarta en el que se representa el porcentaje de aprobados y suspensos predichos.

- Vídeos: Podrá ver gráficos que representan la interacción de los estudiantes con los vídeos del curso, como:
 - o Gráficos de barras con el nº de alumnos que han visto cada vídeo.
 - o Gráfico de “tarta” representando cada acción que los alumnos han realizado con ese vídeo, número de veces que ha rebobinado hacia adelante, hacia atrás y veces que han pulsado el *play* o el *stop*.
 - Comentarios: Podrá ver el número de comentarios realizado por cada alumno y además el número de alumnos que han realizado comentarios. Esta información se representará mediante un gráfico de barras a lo largo del tiempo.
 - Perfil: Podrá ver los alumnos que tienen completo su perfil mediante un gráfico de tarta. A su vez puede ver el número de perfiles completos y el porcentaje de alumnos según tengan completo el perfil (60%, 80%, ...) ya sea mediante el gráfico de tarta o uno de líneas.
 - Accesos a la plataforma: Podrá conocer cuántas veces acceden los alumnos a la plataforma durante el mes, información representada mediante gráficos de barras y de líneas.
 - Actividades: Podrá ver gráficos sobre las distintas actividades y las acciones que se realizan sobre ellas.
 - o Número de actividades realizadas por él a lo largo del curso.
 - o Número de intentos que necesita para la actividad correspondiente.
 - o Si ha finalizado esa actividad con éxito o no.
- Todo ello se visualizará de 3 formas distintas:
- o Un gráfico de áreas que tendrá las 3 variables a lo largo del tiempo, para comprobar la evolución tanto del número de actividades realizadas como del número de intentos para resolver cada una de ellas y del éxito en las mismas.
 - o Un gráfico de barras agrupado por meses, para ver la tendencia que tienen los alumnos a terminar correctamente los ejercicios respecto a los intentos.
 - o Un gráfico de línea con el porcentaje de aciertos de todos los estudiantes a lo largo del tiempo, que representaría en cierto modo la evolución de los estudiantes, reflejada en el éxito del mismo en la realización de las actividades del curso.
- Actividad (acciones generales en el curso): Se podrá ver mediante barras y líneas a lo largo del tiempo, cada interacción que realice un usuario con el curso.

En la siguiente figura se muestra un diagrama de casos de uso donde se puede ver cada acción que puede realizar un usuario, en este caso el profesor. El usuario podrá tanto iniciar sesión, como salir de ella y tendrá dos formas de seguimiento del curso: individual para cada alumno, o grupal de todo el curso. Ambas tendrán los mismos gráficos de visualización exceptuando los gráficos de aprobados ya que solo tienen sentido ver los porcentajes de predicción de aprobados de forma global.

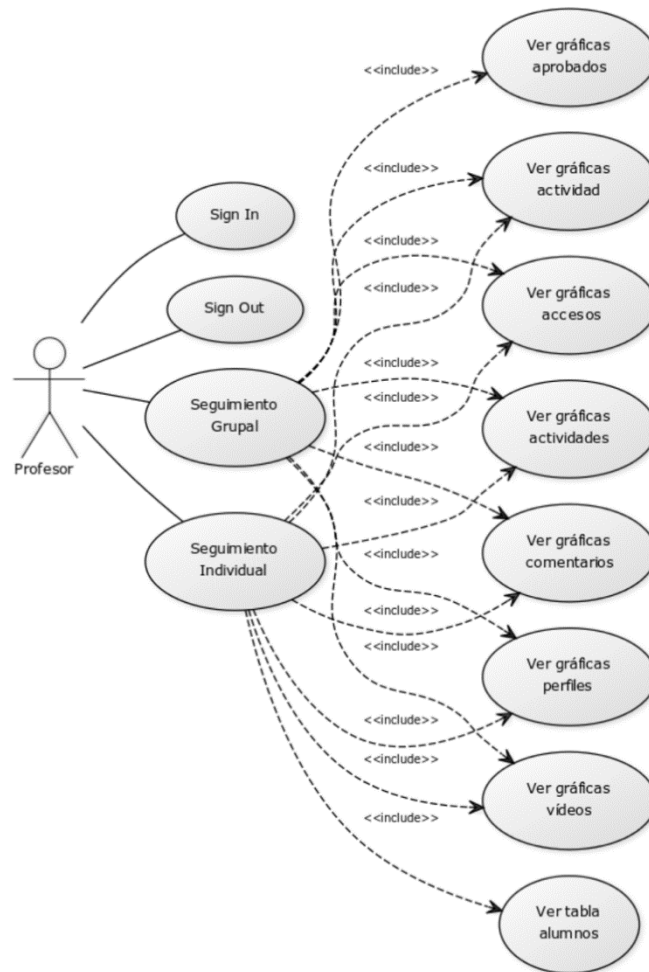


Figura 2: Diagrama de casos de uso

3.5 Propuesta de gráficos

El seguimiento de los alumnos se hace posible gracias a la implementación de gráficos, que usando ciertas variables hace fácil la monitorización. Puesto que en el apartado anterior se han mencionado las formas de visualización, a continuación, se resumen las gráficas propuestas:

- Gráficas sobre aprobados: gráficos de tarta.
- Gráficas sobre actividades: gráficos de área, barras y líneas.
- Gráficas sobre accesos: gráficos de barras y líneas.
- Gráficas sobre videos: gráficos de tarta y de barras.
- Gráficas sobre perfiles: gráficos de tarta.
- Gráficas sobre comentarios: gráficos de barras.
- Gráficas sobre actividad: gráficos de barras.

La novedad en la implementación se encuentra en la tabla *users*. Esto se debe a que el *framework* Laravel contiene una implementación mediante modelos y controladores para el manejo de usuarios en la aplicación, siendo esta tabla la que controla el registro y el acceso a la aplicación. Además, se ha introducido un campo *rol*, que está pensado para que en un futuro la aplicación no esté solo orientada a los profesores.

4 Desarrollo

En esta sección se describe cómo se ha desarrollado la aplicación, indicando las tecnologías utilizadas y pequeños ejemplos de uso de esas tecnologías.

4.1 Tecnologías utilizadas

Debido a que la aplicación es para la Web, los lenguajes utilizados son HTML, PHP y JavaScript. Además, se ha utilizado un cuarto lenguaje (Python) para el tratado de archivos CSV y de los *logs* con los datos reales de los MOOCs y un quinto, SQL, para el uso de la base de datos. A continuación, se hará una breve descripción de estos lenguajes y su uso.

- **HTML (versión 5) [11]**
Hypertext Markup Language, es un lenguaje específico de páginas web, siendo un estándar del Consorcio WWW (World Wide Web). Es el lenguaje estándar utilizado en la visualización de páginas web siendo este lenguaje utilizado por todos los navegadores.
- **PHP (versión 7.1) [12]**
Hypertext Preprocessor, es el lenguaje utilizado en general en el lado del servidor y con uso principal para el desarrollo de páginas web, dando dinamismo a páginas HTML.
- **JAVASCRIPT (versión 10.0) [13]**
JS en abreviado, es un lenguaje orientado a objetos que es utilizado tanto del lado del cliente como del servidor (Node.js), dando dinamismo a las páginas web.
- **PYTHON (versión 2.7) [14]**
Es un lenguaje muy potente y multiplataforma que soporta la orientación a objetos, programación imperativa y funcional.
- **SQL (versión 5.7) [15]**
Es un lenguaje específico de bases de datos, permitiendo la interacción con ella y realizando operaciones y consultas.

Estos lenguajes son los utilizados por los *Frameworks* Laravel y Bootstrap, que son los entornos que se han utilizado en el desarrollo de esta aplicación. A continuación, se describen los motivos por los que se han utilizado estos *Frameworks*.

- **Laravel (versión 5.4) [16]**
La utilización de este Framework se debe a los siguientes motivos:
 - La sencillez en el uso de rutas y enlaces.
 - Contiene una herramienta de interfaces bajo líneas de comando llamada Artisan, que permite lanzar el servidor, crear clases y controladores, realizar migraciones a la base de datos, ...
 - Es muy modular, tiene un sistema de paquetes y drivers que extienden la funcionalidad.
 - Permite el uso del Framework Bootstrap.
 - Tiene un sistema de plantillas Blade, que aparte de ser más simples respecto al código, son más rápidas y con mejor rendimiento.

- Tiene una librería interna que permite realizar la validación de datos más sencilla.
 - Permite el uso de casi todas las bases de datos, además de hacer muy sencillo tener la base de datos sincronizada mediante migraciones.
 - Se tiene una gran cantidad de documentación sobre Laravel.
- Bootstrap (versión 3.7)
La utilización del Framework Bootstrap se debe a las siguientes razones:
 - Permite una fácil integración con librerías de JavaScript.
 - Facilita el diseño de interfaces además de tener una gran cantidad de plantillas gratuitas.
 - La plantilla gratuita sb admin 2, permite la implementación de las gráficas que se necesitan en este proyecto.
 - Se puede introducir dentro del Framework Laravel.

Se puede obtener una gran cantidad de tutoriales y documentación.

Pese a que la versión de Bootstrap utilizada contiene iconos, se ha preferido utilizar una fuente de iconos gratuita:

- Fontawesome (versión 4.7)
Consiste en una fuente con una gran cantidad de iconos, que es muy fácil de incorporar en el código. Para su incorporación es necesario descargarlo, introducirlo dentro de la carpeta del proyecto y finalmente referenciarlo en el archivo css.

Finalmente se nombran los editores de texto utilizados para el desarrollo y la base de datos:

- PhpStorm: Plataforma utilizada para el desarrollo de la aplicación web.
- SequelPro: Plataforma utilizada para la creación de la base de datos y sus posteriores modificaciones.
- Pycharm: Plataforma utilizada para llevar los datos reales almacenados en los logs a la base de datos.

4.2 Programación

4.2.1 Vistas

Lo primero que se hizo fue la creación de unos bocetos a mano para tener una primera vista de cómo sería la aplicación. A continuación, en la siguiente imagen se muestra un ejemplo de estos bocetos. En ella se puede observar un primer vistazo de cómo van a ser las pantallas en general. Estas, estarán formadas por una barra de navegación en la parte superior y un menú lateral. En el resto de la pantalla se mostrarán las distintas gráficas para realizar la monitorización. [17]

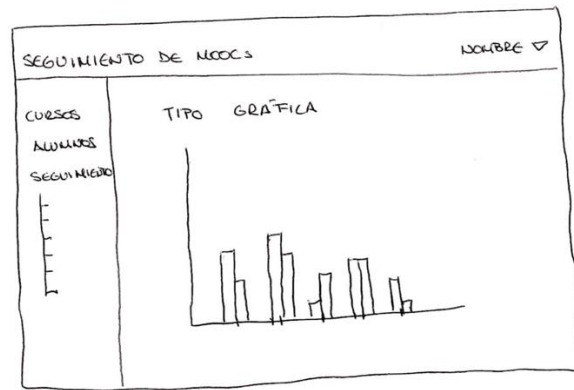


Figura 4: Boceto de una pantalla

Después de tener claro cómo va a ser la interfaz de la aplicación, se ha procedido a la codificación de estos bocetos. Durante el desarrollo de estos, no ha sido necesaria la realización de modelos, solamente las vistas. Sin embargo, se ha implementado un controlador básico cuya única función es servir la vista al usuario, sin datos.

Estas vistas están desarrolladas enteramente en el lenguaje HTML con los gráficos en JavaScript, con datos estáticos de prueba. Aquí es donde entra la parte de Bootstrap, utilizando sus plantillas para una visualización más limpia y bonita. Además, para la generación de gráficos se ha utilizado la plantilla de Bootstrap “SB Admin 2”, que permite introducir las gráficos de forma gratuita. También se aprovecha esta plantilla para introducir otras herramientas como son botones, barras de navegación, ...

4.2.2 Base de datos de prueba

Para la creación de la base de datos de prueba lo primero que se ha tenido que hacer es un estudio detallado de los *logs* y archivos con los datos de los cursos, el registro de todas las acciones realizadas por los usuarios y los datos adicionales generados a partir de ese registro [17]. Este estudio tiene el fin de identificar las variables que van a ser útiles para la creación de los gráficos.

Una vez realizado este estudio, se procede a la implementación de una base de datos de prueba con datos ficticios. De esta forma se podrá agilizar la comprobación de si la creación de la base de datos ha sido correcta y hacer cambios más rápidamente que si nos paramos a obtener e incorporar los datos reales a partir de los ficheros mencionados.

4.2.3 Páginas dinámicas

Durante esta fase del desarrollo de la aplicación entra en juego el uso de los modelos. Los modelos permiten acceder a la base de datos para realizar las consultas pertinentes y así servir los datos que se requieran. Estos modelos a veces tienen funciones para relacionar unas tablas con otras o para servir los datos en un formato.

El controlador llama al modelo y realiza las consultas necesarias para obtener todos los datos que sea necesario mostrar en la vista. Laravel contiene clases, con una sintaxis que permite

realizar las consultas a la base de datos fácilmente. Estas consultas son seguras, en el sentido de que evitan posibles ataques de inyección de código SQL.

A continuación, se pueden ver fragmentos de código del modelo, vista y controlador para mostrar el funcionamiento del *Framework* de Laravel. El código permite mostrar los datos de los alumnos en una tabla junto a su predicción.

MODELO

En la siguiente imagen, se puede ver cómo el modelo accede a la tabla de alumnos. También tenemos la función *aprobado*, que relaciona la tabla de alumnos con la de aprobados (ruta del modelo aprobado) a través de su clave primaria y foránea.

```
<?php
namespace App\Modelos\Basicos;
use Illuminate\Database\Eloquent\Model;

class Alumno extends Model
{
    protected $table = 'alumnos';

    public function aprobado(){
        return $this->hasOne( 'related: 'App\Modelos\Graficas\Aprobado', foreignKey: 'user_id', localKey: 'user_id');
    }
}
```

Figura 5: Modelo Alumno

CONTROLADOR

La función del controlador de la figura 5 es acceder a los datos de los alumnos y devolverlos junto a la vista.

Lo primero que comprueba el controlador es que el usuario ha iniciado sesión, ya que si no es así no podrá acceder a la página. Si se ha intentado acceder sin iniciar sesión, la página redirigirá a una respuesta en *json* de No autorizado.

En caso de que el usuario haya realizado el inicio de sesión, el controlador obtiene sus datos. Con los datos del usuario podrá realizar la búsqueda de datos en los modelos de alumnos. Como se puede ver, se realizan consultas al estilo Laravel en los modelos para obtener todos los datos de los alumnos agrupados por 'id'. Finalmente, todos los datos de alumnos obtenidos son devueltos junto a los datos del profesor y la vista.


```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Modelos\Basicos\{Profesor, Alumno, Profesor_curso};
use Auth;

class Controller_alumno extends Controller
{
    public function alumno($curso_id){

        $prof = Auth::user();
        if(!$prof)
        {
            return response()->json( data: "No autorizado", status: 403);
        }
        $profesor = Profesor::where( column: 'id', $prof->profesor->id)->get()->first();
        $alumnos = Alumno::where( column: 'curso_id', $curso_id)->get();

        $datos =[
            'profesor' => $profesor,
            'alumnos' => $alumnos
        ];
        return view( view: 'alumnos.alumnos')->with('datos',$datos);
    }
}

```

Figura 6: Controlador Alumnos

VISTA

En este punto la vista ya no solo contiene código HTML, ya que para mostrar todos los datos servidos por el controlador hace falta usar PHP. Esta vista tendrá una extensión Blade. Esta plantilla permite una gran modularidad, haciendo que cada archivo .blade tenga la menor cantidad de código posible. De esta forma podemos evitar repetir código innecesariamente, utilizando plantillas que el motor Blade de Laravel se encargará de ensamblar. A continuación, podemos ver un ejemplo de la plantilla principal, donde se va ensamblando el código. En la foto se visualiza cómo incluimos otras páginas dentro de la plantilla.

```

<body>
<div id="wrapper">
    <!-- Navigation -->
    @include('layouts.navbar')

    @section('content')
    @show
    </div>
    <!-- /#wrapper -->
</body>

```

Figura 7: Plantilla principal

Finalmente, en la siguiente imagen se observa cómo introducimos los datos mediante PHP dentro de la vista para finalmente mostrarlos.

```

<tbody>
@foreach($datos['alumnos'] as $alumno)

    <tr>
        <td>{{ $alumno['user_id'] }}</td>
        <td>{{ $alumno['language'] }}</td>
        <td>{{ $alumno['location'] }}</td>
        <td>{{ $alumno['gender'] }}</td>
        <td>{{ $alumno['year_of_birth'] }}</td>
        <td>{{ $alumno['level_of_education'] }}</td>
        <td>{{ $alumno['goals'] }}</td>
        <td>{{ $alumno['bio'] }}</td>
        <td>@if($alumno['profile_image_uploaded_at'])
            s
        @else
            n
        @endif
    </td>
    @if($alumno->aprobado['grade'] >= 0.5)
        <td class="success">{{ $alumno->aprobado['grade'] }}</td>
    @else
        <td class="danger">{{ $alumno->aprobado['grade'] }}</td>
    @endif
    <td>
        <a href="{{ route('seguimiento.alumno', $alumno['user_id']) }}">
            <i class="fa fa-arrow-right" aria-hidden="true"></i>
        </a>
    </td>
    </tr>
@endforeach

```

Figura 8: Contenido de la vista de alumnos

En la figura 6 se muestra la plantilla donde aparece el código `@section('nombre_seccion')`. Ahí es donde se introduce nuestro código de la figura 7. Por motivos de espacio, solo se muestran las líneas de código que visualizan las variables pasadas por el controlador.

Para que esta vista se introduzca dentro de la plantilla tiene que ser declarada la plantilla mediante `@extends('nombre_de_la_plantilla')` y creada entre los trozos de código `@section('nombre_seccion')` y `@endsection`

A continuación, se muestra un diagrama de secuencia perteneciente a la acción de mostrar los aprobados de los alumnos, para que se entienda el funcionamiento del MCV.

Estando en la vista de seguimiento, devuelta por el controlador de seguimiento, el profesor hace *click* en el botón de aprobados. A continuación, se realizará una llamada a la función `aprobados`. En consecuencia, el controlador de seguimiento realiza las llamadas a los modelos `profesor`, `profesor_curso` y `aprobado`, que realizan las consultas pertinentes a la base

de datos devolviendo los datos necesarios junto a la vista final de aprobados.

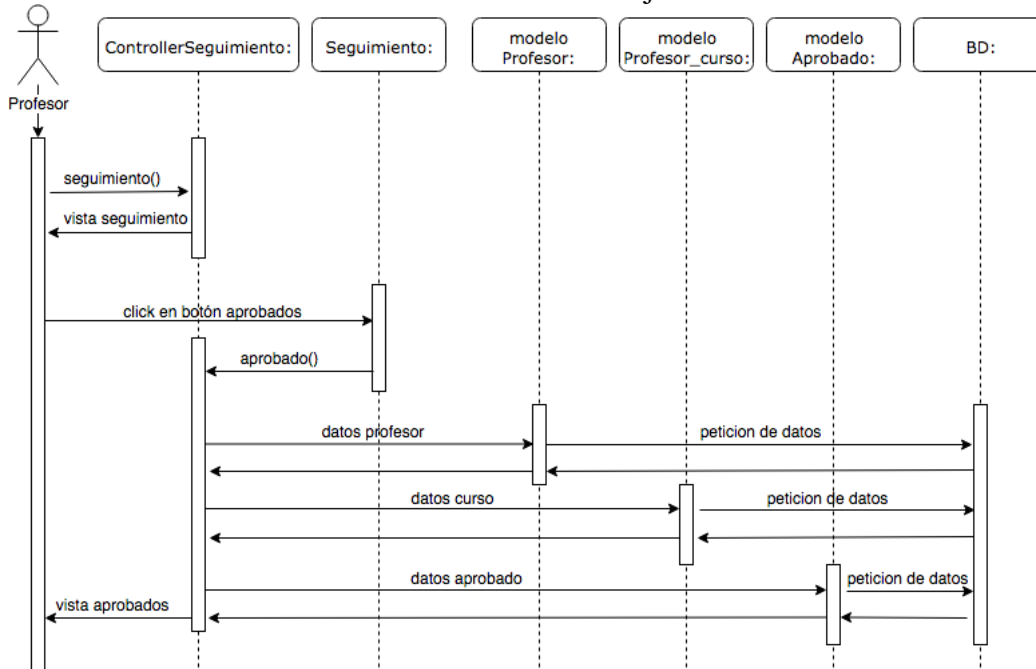


Figura 9: Diagrama de secuencia de la vista aprobados

4.2.4 Datos reales

Por último, una vez que ya se ha probado que la aplicación muestra todos los datos de prueba de la base de datos correctamente, se procede a la introducción de datos reales.

Para ello se utilizan los ficheros y eventos descritos en la sección 3.2. Para tratar estos ficheros se ha utilizado un script creado en Python que trata estos datos quedándose con los de interés, para finalmente introducirlos en la base de datos. Este proceso es explicado detalladamente en la sección 5, en la parte de integración.

4.2.5 Rutas y accesos

Todo lo anterior no funcionará si no se tienen definidas unas rutas, a través de las cuales accedemos a cada página de la aplicación web. En Laravel, las rutas se definen en un fichero llamado *web.php*. Las rutas no solo indican la URL a la que dirigirse, también el método utilizado siendo estos métodos GET y POST. Además, Laravel tiene una forma fácil de encadenar rutas como se puede ver en la imagen siguiente. En ella vemos las rutas necesarias para acceder a la vista de alumnos y la página de seguimiento detallado del alumno.

```
Route::prefix('/visualizacion')->group(function () {
    Route::get('/alumnos', 'Controller_alumno@alumno');
})
Route::prefix('/seguimiento_alumno')->group(function () {
    Route::get('/{alumno_id}', 'Controller_seguimiento@seguimiento_alumno')->name('seguimiento.alumno');
});
```

Figura 10: Rutas de alumnos y seguimiento detallado

En la figura anterior se muestra un ejemplo de ruta para acceder al seguimiento del alumno. Primero se crea un prefijo */visualización*, evitando así se repita el mismo código en todas las rutas pertenecientes a la visualización. Después se declara la ruta para acceder a la tabla con los alumnos que, recopilando lo anterior, sería */visualización/alumnos*. Esta ruta llama al controlador de alumno, concretamente a la función que devuelve la vista y los datos pertinentes. Finalmente se vuelve a hacer otra agrupación, dentro de la cual se llama a la vista del seguimiento detallado del alumno. En este caso la ruta llevará el id del alumno que obtendrá de un formulario de tipo GET.

5 Integración, pruebas y resultados

5.1 Integración

A continuación, se describe la introducción de los datos pertenecientes a los alumnos en la base de datos. Los datos, como ya se ha mencionado anteriormente, se extraen de los ficheros csv y logs.

Para realizar el volcado del csv en la base de datos, en un principio se han limpiado filas y columnas en blanco. Además, se han eliminado campos repetitivos o que se han considerado de poco interés para esta aplicación, como por ejemplo el curso, que es el mismo para todos, o el país y la ciudad del alumno que ya viene mencionado en localización.

Los ficheros de los que proceden estos datos son los *Certificate*, *Profile* y *Social* de cada curso. Una vez que se han tratado estos csv, se ha procedido a introducir los datos en la base de datos. Para ello se han usado las librerías de Python *MySQLdb* y *csv*.

En cuanto al trabajo realizado con los *logs*, ha sido más tedioso. Se ha tenido que realizar un script en Python para tratar los datos de estos eventos para después meterlos en la base de datos. El script recorre todos los ficheros de eventos leyendo uno a uno y extrayendo las variables más significativas, que son: *user_id* (id del usuario responsable del evento), *course_id* (id del curso en el que se ha generado el evento), *event_type* (tipo de evento que se ha generado pudiendo ser acciones con libros, videos, navegación en la página y acciones con problemas), *time* (momento en el que se ha generado el evento). Si los eventos son de tipo vídeo o de problemas, también registrará el id de dicho vídeo o problema.

Finalmente, una vez ordenadas estas variables, dependiendo del tipo que sean se introducen en sus respectivas tablas de la base de datos.

5.2 Pruebas

La fase de pruebas consiste en probar que el funcionamiento de la aplicación web es correcto. Para ello se han realizado pruebas tanto unitarias como de aplicación para garantizarlo. Para que sea más sencilla su realización y ejecución se ha utilizado el *framework* de Laravel ya que trae incluido *PHPUnit* (entorno de PHP cuya función es la realización de pruebas unitarias) [17]. Este entorno realiza *assertions* para probar el correcto funcionamiento de la aplicación. También aprovechamos otra función de Laravel, el paquete *Integrated test package*, cuya función es la realización de las pruebas de aplicación.

Llegados a este punto voy a aprovechar para plantear la siguiente pregunta: ¿Qué diferencian las pruebas unitarias a las de aplicación? Las pruebas unitarias se dedican a comprobar el correcto funcionamiento de las funciones “internas” de la aplicación, mientras que las pruebas de aplicación se centran en probar lo que el usuario ve [18].

Aunque parezca que lo primordial es que el usuario pueda visualizar las páginas de forma correcta, es decir que la aplicación pase las pruebas de aplicación, las pruebas unitarias son necesarias por las siguientes ventajas:

- Son más fáciles de depurar que las pruebas de aplicación, en caso de que la prueba falle.

- Cubren muchos más casos que las pruebas de aplicación, por lo que permiten comprobar más escenarios que si se realizaran solo pruebas de aplicación.
- Son más rápidas que las pruebas de aplicación. Esto se debe a que computacionalmente es más rápido cargar una función que toda una página.

A continuación, se muestran dos ejemplos de las pruebas realizadas con una breve descripción de ambas.

```
/** @test */
function load_routes()
{
    $this->get( uri: '/seguimiento/aprobados' )
    ->assertStatus( status: 200);
}
```

Figura 11: Ejemplo de prueba unitaria

```
/** @test */
function login()
{
    $usuario = new User(['name' => 'Prueba',
        'email' => 'prueba@email.com',
        'password' => bcrypt( value: '1234'),
        'rol' => 'profesor',
        'estado' => 0]);

    $p = auth()->login($usuario);
    $p->assertStatus(200);
}
```

Figura 12: Ejemplo de prueba de aplicación

5.3 Resultados

En este apartado se van a mostrar los resultados finales tras el diseño y la integración de los datos mediante unas capturas de pantalla de la aplicación, seguidas de una pequeña descripción del funcionamiento de dicha página. Puesto que la aplicación consta de varias páginas de diseño, se pondrán únicamente dos ejemplos, pudiendo ver el resto de las capturas de la aplicación en los anexos.

1. Pantalla de inicio de sesión

El usuario introducirá su correo y contraseña para entrar en la aplicación. Habrá tenido que ser registrado previamente para poder acceder. Si se selecciona la opción “recordarme”, la aplicación recordará la dirección de correo y la contraseña. También se tendrá la opción de cambiar la contraseña en el caso de haberla olvidado. En ese caso se enviará un formulario por email al correo introducido, para que el usuario introduzca la nueva contraseña.

The login screen features a light gray header with the word "Login". Below the header, there are two input fields: "Dirección de correo" (Email) and "Contraseña" (Password). A checkbox labeled "Recordarme" (Remember me) is positioned below the password field. At the bottom, there is a blue "Login" button and a link that says "He olvidado mi contraseña" (I forgot my password).

Figura 13: Pantalla de inicio de sesión

2. Elección de curso

Tras haber entrado en la aplicación, el profesor podrá elegir entre sus cursos impartidos. Estos cursos le aparecerán mediante botones en el centro de la pantalla, que adaptarán su posición y tamaño a la interfaz en la que se muestren.

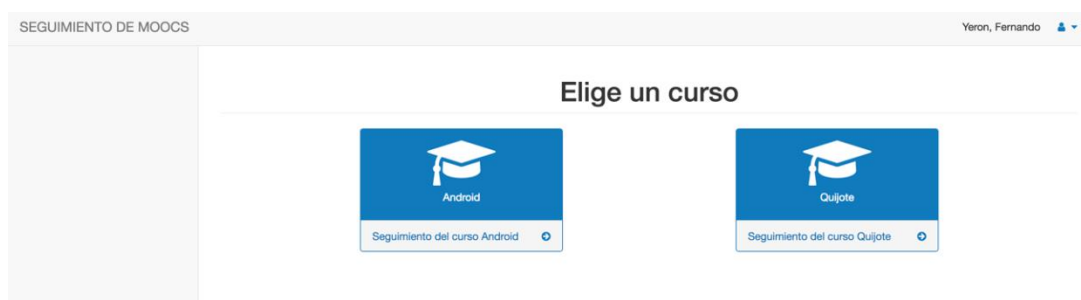


Figura 14: Pantalla elección de curso

3. Pantalla principal

Esta es la página principal, donde se podrá acceder a las funciones tanto desde la barra lateral como desde los botones que se encuentran en el centro de la página. La barra lateral tiene 4 elementos. El primero es home, que redirige a esta página, siendo más fácil volver a la página principal en caso de estar en otra. La siguiente es la de curso, que redirige a la página descrita anterior en caso de querer cambiar el seguimiento a otro curso.

El botón de alumnos permite ir a la vista de los alumnos con sus datos (se puede acceder a dicha página tanto mediante la barra lateral como desde el botón central). Al pulsar en la barra lateral el seguimiento, este se desplegará dando todas las opciones disponibles. Los datos seguimiento mostrados en esta opción serán los generales, de todo el curso, ya que si se quiere acceder a una monitorización individual se debe hacer desde la opción del menú Alumnos (se ofrecerán más detalles sobre esto en el punto 6 de esta sección). Finalmente, en caso de querer salir de la aplicación se pulsará en el icono situado al lado del nombre del usuario en la barra superior, dando la opción de cerrar sesión.

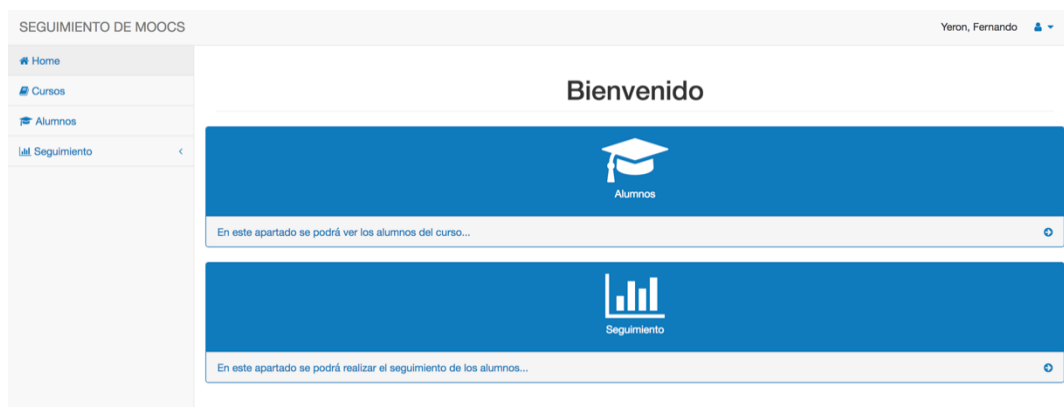


Figura 15: Pantalla principal

4. Pantalla de seguimiento

Desde esta pantalla se puede acceder a varios tipos de seguimiento del curso. Como se ha mencionado en el apartado anterior, los datos son generales. Las opciones para la visualización de datos son: gráficos de aprobados, actividades, accesos, acciones sobre los vídeos, actividad del usuario en general y gráficas de perfiles y comentarios.



Figura 16: Pantalla de seguimiento

5. Tabla de alumnos

Como se muestra en la figura 18, la tabla de alumnos incluye todos los datos que se tienen sobre los alumnos, así como la previsión de aprobados o suspensos. Esta tabla permite filtrados por columnas y búsquedas por nombre.

También se permite modificar la tabla, añadiendo entradas para que se muestren más datos, o al revés, para mostrar menos entradas. Los botones inferiores de la derecha permiten navegar por las distintas tablas de la página.

El icono de la flecha de cada entrada de la tabla permite acceder a los datos de seguimiento más detallados de dicho alumno.

Finalmente aparecen unos *checkboxes*, que permiten la selección de dos alumnos, dando así la posibilidad de comparar los datos detallados de esos alumnos, para poder comparar su evolución o sus resultados.

Alumnos en MOOCS

Datos de los alumnos

Show 10 entries

ID usuario	Idioma	Localización	Género	Año nacimiento	N. educación	Metas	Biografía	Foto perfil	Prevision
94246	Español	Yerba Buena - Tucumán - Argentina						n	0.92
94157	Spanish	Santiago Chile	m	1959	m			n	0
94030	Spanish	Cochabamba, Bolivia						n	0.01
93882	English	Greece	m	1981	m			n	0.35
93865	Spanish	Bayamon, Puerto Rico						n	0
93168	Spanish	Madrid						n	0.01
93159	spanish english french catalan	Ampost						n	0
92975	Español	Bolivia	m	1996	b			n	0.01
92888		Peru						n	0
92680	Spanish or English	Beruta, Miranda, Venezuela						n	0

Showing 1 to 10 of 500 entries

Previous 1 2 3 4 5 ... 50 Next

[Compartir alumnos](#)

Figura 17: Pantalla de datos de alumnos

A continuación, se muestran dos ejemplos de pantallas de seguimiento a nivel global: aprobados y perfiles. El resto de las pantallas de seguimiento se mostrarán en los anexos.

6. Porcentaje de aprobados

Muestra un gráfico con el porcentaje de aprobados y suspensos del total de alumnos del curso, con forma de “queso” o “tarta”. Si se pasa el cursor del ratón sobre el gráfico aparecerá el porcentaje exacto de esa zona, como se puede ver en la imagen.



Figura 18: Pantalla de porcentaje de aprobados

7. Ejercicios

Esta página muestra información sobre los ejercicios realizados por todos los alumnos durante el curso, facilitando así el seguimiento de este aspecto. Las gráficas

muestran los datos sobre la cantidad de ejercicios realizados a lo largo del tiempo, la cantidad de aciertos que se han producido y los intentos realizados. Se han implementado tres gráficos distintos, de entre los cuales destaco el de la figura 19. En él se puede diferenciar tres áreas distintas a lo largo del tiempo. La cantidad de ejercicios propuestos a los estudiantes (a los que estos acceden) se muestran en color azul, el número de intentos que realizan para resolverlos en gris y el número de aciertos en verde. Con este tipo de gráfica puede detectarse rápidamente si se han realizado muchos intentos y si se ha acertado o no. Esta misma información se muestra con distintos tipos de gráficos, para facilitar la visualización de los mismos datos. En el anexo se muestran otras gráficas diferentes.

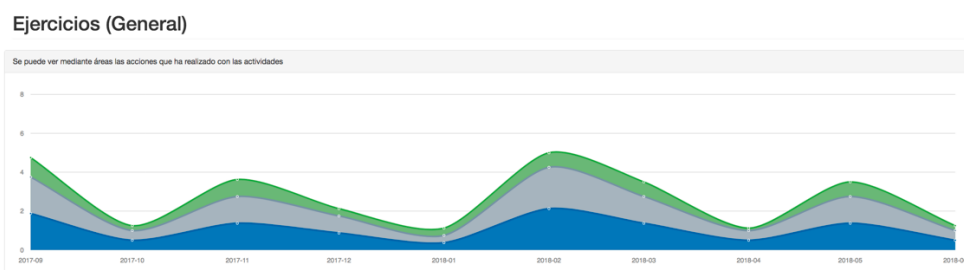


Figura 19: Ejercicios general

También es posible ver la evolución de cada estudiante individualmente a lo largo del curso, por ejemplo, en cuanto a número de ejercicios realizados y éxito en ellos. La figura 20 muestra el mismo tipo de información que la 19, pero en este caso para un estudiante concreto. Como se puede observar, este alumno en un inicio empezó resolviendo correctamente los ejercicios propuestos, pero sin embargo después accedió a muchos menos ejercicios, y los que realizó los hizo de manera incorrecta.

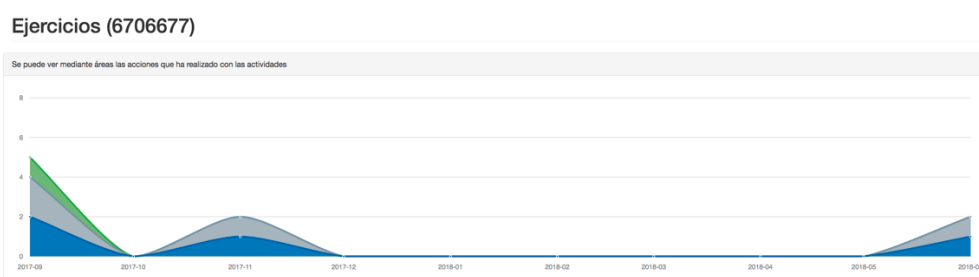


Figura 20: Ejercicios individual

8. Accesos al MOOC

La finalidad de esta pantalla es poder mostrar el número de accesos de los alumnos al MOOC a lo largo del tiempo. Para ello se generan dos tipos de gráficos, que se muestran a continuación.

En la figura 21 se pueden ver los dos tipos de gráficas. La primera de ellas, un gráfico de barras dividido en meses, muestra la cantidad de alumnos que acceden a la plataforma del MOOC. También se muestra con una gráfica de líneas, para ofrecer una vista distinta, que puede gustar más o ser más útil para algunos docentes, mientras que otros pueden preferir los gráficos de barras. En esta última gráfica se

puede ver, por ejemplo, que durante este curso Enero es el mes en el que más accesos se han realizado.

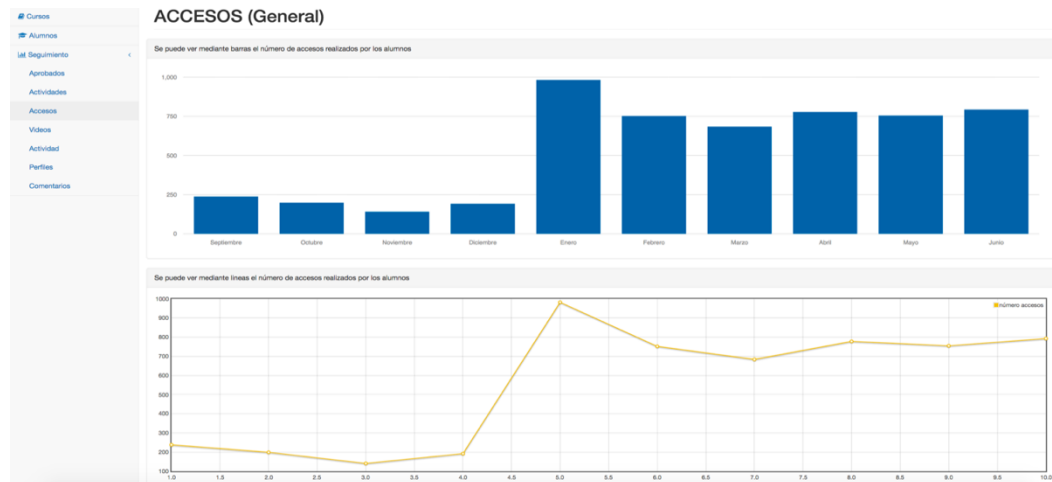


Figura 21: Accesos general

9. Selección de seguimiento de dos alumnos

A esta página se accede desde la tabla de alumnos. Para ello se seleccionan dos de los alumnos que se quieren comparar y se pulsa en siguiente. La pantalla a la que se accederá será exactamente igual que la de seguimiento del apartado 5.

A diferencia de los seguimientos generales o individuales, estos serán comparativos. Como ejemplo se puede ver la figura 22, donde se muestra la comparación de los accesos al MOOC de dos alumnos, cada uno representado de un color. De esta forma un profesor podría comparar en este caso a cualquier estudiante con el alumno con mayor número de accesos.

ACCESOS (210, 14411)

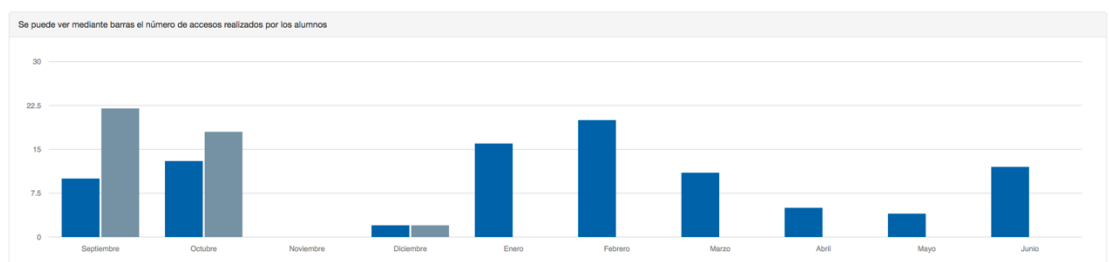


Figura 22: Comparativa de accesos

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este trabajo se ha realizado el análisis, diseño e implementación de una aplicación web desde cero. El objetivo de esta aplicación es que el docente de un curso MOOC pueda llevar a cabo el seguimiento y monitorización de los alumnos que están realizando ese curso, y de esta forma pueda saber cómo va cada uno de ellos, cómo va el conjunto de alumnos en general y localizar situaciones de riesgo o indicios de que algún estudiante probablemente vaya a suspender.

La creación de esta aplicación viene dada por la alta demanda que hay en estos cursos, que actualmente cuentan con una gran cantidad de estudiantes que dificulta su seguimiento por parte de los profesores. La tasa de suspensos en los MOOCs es bastante alta y es interesante proporcionar a los profesores herramientas para tratar de detectar las causas, que pueden ser variopintas; por ejemplo, habrá estudiantes que solo “pasaban por allí” y querían echar un vistazo a los contenidos del curso, estudiantes que vieran los vídeos y consultaran el material, pero no realizaran ejercicios o estudiantes que, pese a intentarlo, no los realizaron correctamente. La exploración de forma visual de las acciones de los estudiantes con los cursos puede ayudar a los profesores a realizar un análisis del desempeño de los estudiantes en este tipo de cursos o incluso detectar cuestiones mejorables en los mismos (por ejemplo, si un gran número de estudiantes fallan en unos determinados ejercicios, puede indicar que los contenidos relacionados no están claros o incluso que puede haber algún fallo en esos ejercicios).

Se han cumplido con los objetivos planteados:

1. Se han implementado unas vistas estáticas para la aplicación.
2. Se ha probado el funcionamiento con una base de datos de prueba, con datos ficticios.
3. Se han introducido elementos dinámicos a las vistas anteriores.
4. Se han introducido los datos reales en la base de datos. Se ha comprobado el correcto funcionamiento con datos reales.

Este trabajo me ha permitido poner en práctica conocimientos adquiridos durante el grado, como realizar la documentación y descripción de un proyecto, desarrollo web y programación en distintos lenguajes utilizados en el grado. Además, he aprendido muchos otros como el uso del *framework* de Laravel y Bootstrap y verme involucrado en un gran proyecto desde cero yo solo.

La aplicación está funcionando en la actualidad y queda a disposición de los docentes para su utilización. En el siguiente apartado se indican algunas sugerencias de mejora.

6.2 Trabajo futuro

Como trabajo futuro se plantean las siguientes ideas.

En primer lugar, se podría realizar una ampliación de usuarios en la aplicación. Esta API está orientada solo a usuarios de tipo docente, para que puedan realizar el seguimiento del alumnado. También podría orientarse al perfil del alumno, de forma que el alumno pueda ver el desarrollo de su curso y la predicción de éxito o fracaso del curso en cada momento.

Otra mejora interesante sería introducir un chat o enlace al e-mail para que el profesor pueda contactar fácilmente con el alumno que elija, o incluso realizar grupos con varios. De esta forma el profesor no solo podrá ver quienes van mal encaminados en el curso, sino que también podrá intentar poner remedio a ello contactando con los estudiantes concretos a la vista de información específica (sería interesante, por ejemplo, facilitar que el profesor pudiera adjuntar algún gráfico concreto en su mensaje, si lo considerara oportuno).

Finalmente, se podría investigar la posibilidad de incorporar este software para la visualización de datos y seguimiento de los alumnos en los propios MOOCs, si fuera posible. Por ejemplo, se podría explorar esta posibilidad con los MOOCs en EdX de la UAM. Si a esto se uniera la incorporación de un módulo de predicción de éxito o fracaso (por ejemplo, basado en el trabajo de [17]), cuyos datos se visualicen a través de la herramienta desarrollada en este TFG, se podrían enriquecer bastante este tipo de cursos, desde el punto de vista tanto de los docentes como de los estudiantes.

Referencias

- [1] L. G. Aretio, «Contextos universitarios mediados,» 8 Noviembre 2013. [En línea]. Available: <https://aretio.hypotheses.org/918>. [Último acceso: 22 Junio 2018].
- [2] R. d. Río, «Los MOOC y la siguiente generación,» 3 Junio 2013. [En línea]. Available: <http://blog.catedratelefonica.deusto.es/moocs-y-la-siguiente-generacion/>. [Último acceso: 22 Junio 2018].
- [3] A. P. Muñoz, «Blog de economía de la ALdEA Global,» 16 Octubre 2012. [En línea]. Available: <http://blog.catedratelefonica.deusto.es/moocs-y-la-siguiente-generacion>. [Último acceso: 22 Junio 2018].
- [4] «uc3m,» [En línea]. Available: <http://www.it.uc3m.es/pedmume/ANALYSE/>. [Último acceso: 22 Junio 2018].
- [5] D. J. G. R. Gonzalez y M. F. S. Sarasty, «ResearchGate,» Julio 2017. [En línea]. Available: https://www.researchgate.net/publication/320774591_Estrategia_de_seguimiento_a_las_actividades_de_aprendizaje_de_los_estudiantes_en_cursos_en_linea_masivos_y_privados_MPOC_con_reconocimiento_academico_en_la_Universidad_del_Cauca. [Último acceso: 22 Junio 2018].
- [6] D. V. Sancho T, «Universitat Pompeu Fabra,» 2014. [En línea]. Available: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwido8S7jeLbAhXG0RQKHRWKC98QFjACegQIARA6&url=http%3A%2F%2Fwww.upf.edu%2Fes%2Fweb%2Fusquid-etic%2F-%2Festrategia-pel-seguiment-i-avaluacio-dels-aprenentatges-en-un-moo>. [Último acceso: 22 Junio 2018].
- [7] U. Hernandez, «Códigofacilito,» [En línea]. Available: [29] <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>. [Último acceso: 22 Junio 2018].
- [8] F. Gaitán, «Fernando Gaitán,» 12 Noviembre 2013. [En línea]. Available: [30] <https://fernando-gaitan.com.ar/laravel-parte-3-modelo-vista-controlador/>. [Último acceso: 22 Junio 2018].
- [9] J. A. Camarasa, «Clavei,» 14 Mayo 2012. [En línea]. Available: <https://www.clavei.es/blog/que-aporta-mvc-al-desarrollo-de-aplicaciones-web/>. [Último acceso: 22 Junio 2018].
- [10] «TechTarget,» Enero 2015. [En línea]. Available: <https://searchdatacenter.techtarget.com/es/definicion/Base-de-datos-relacional>. [Último acceso: 22 Junio 2018].
- [11] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/HTML>. [Último acceso: 22 Junio 2018].
- [12] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/PHP>. [Último acceso: 22 Junio 2018].
- [13] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>. [Último acceso: 22 Junio 2018].
- [14] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/Python>. [Último acceso: 22 Junio 2018].

- [15] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/SQL>. [Último acceso: 22 Junio 2018].
- [16] «Gitbooks,» [En línea]. Available: <https://richos.gitbooks.io/laravel-5/content/>. [Último acceso: 22 Junio 2018].
- [17] C. I. Estradas, «Repositorio Uam,» Septiembre 2017. [En línea]. Available: https://repositorio.uam.es/bitstream/handle/10486/681085/Isidro_Estradas_Cristina_tfm.pdf?sequence=1. [Último acceso: 20 Junio 2018].
- [18] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/PHPUnit>. [Último acceso: 22 Junio 2018].
- [19] D. Palacios, «Styde,» 29 Enero 2016. [En línea]. Available: <https://styde.net/como-escribir-pruebas-unitarias-y-de-aplicacion-tipos-diferencias/>. [Último acceso: 22 Junio 2018].
- [20] «Laravel,» [En línea]. Available: <https://laravel.com/>. [Último acceso: 20 Junio 2018].
- [21] «Bootstrap,» [En línea]. Available: <https://getbootstrap.com/>. [Último acceso: 22 Junio 2018].
- [22] «StartBootstrap,» [En línea]. Available: <https://startbootstrap.com/template-overviews/sb-admin-2/>. [Último acceso: 22 Junio 2018].
- [23] «Stackoverflow,» [En línea]. Available: <https://es.stackoverflow.com/>. [Último acceso: 22 Junio 2018].
- [24] «Fontawesome,» [En línea]. Available: <https://fontawesome.com/>. [Último acceso: 22 Junio 2018].
- [25] «W3schools,» [En línea]. Available: <https://www.w3schools.com/>. [Último acceso: 22 Junio 2018].
- [26] «yUML,» [En línea]. Available: <https://yuml.me/diagram/nofunky/usecase/draw>. [Último acceso: 22 Junio 2018].
- [27] «Draw,» [En línea]. Available: <https://www.draw.io/>. [Último acceso: 22 Junio 2018].

Glosario

API	Application Programming Interface
TFG	Trabajo de Fin de Grado
TFM	Trabajo de Fin de Máster
RF	Requisito Funcional
RNF	Requisito no Funcional
CSS	Cascading Stylesheets (lenguaje de estilos)
MCV	Modelo Vista Controlador
CSV	Comma Separated Values (formato de representación de tablas)
Interfaz	Vista de la aplicación
Framework	Entorno de trabajo

Anexos

A Capturas de la aplicación

SEGUIMIENTO DE MOOCS

Inicio Sesión

Figura 23: Pantalla principal

Reset Password

E-Mail Address

Send Password Reset Link

Figura 24: Reseteo de contraseña

Register

Name

E-Mail Address

Password

Confirm Password

Register

Figura 25: Registro solo para administradores



Figura 26: Seguimiento individual

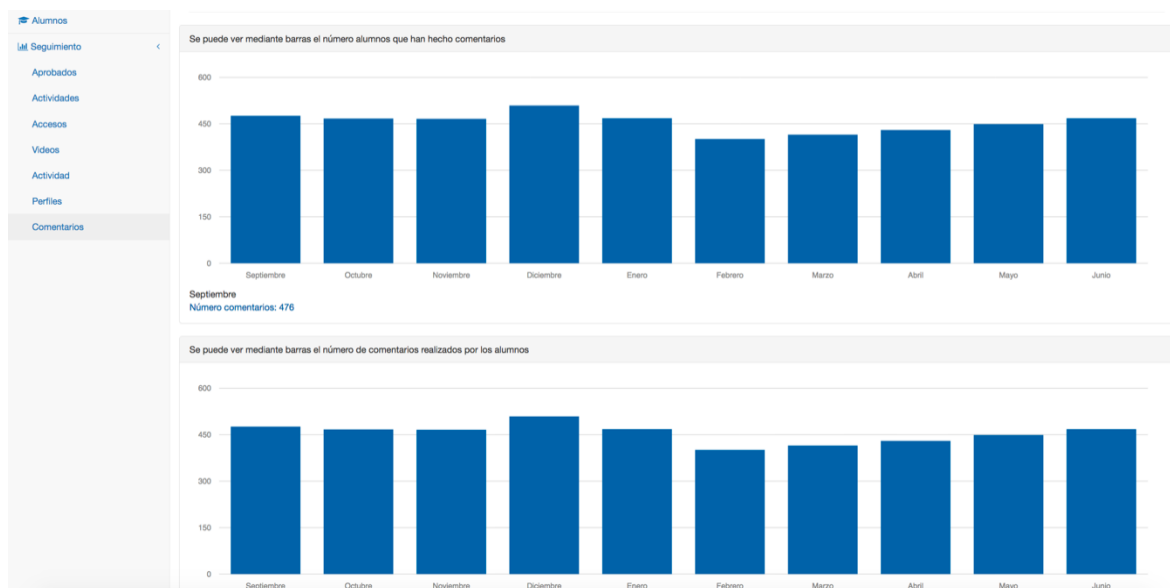


Figura 27: Comentarios general

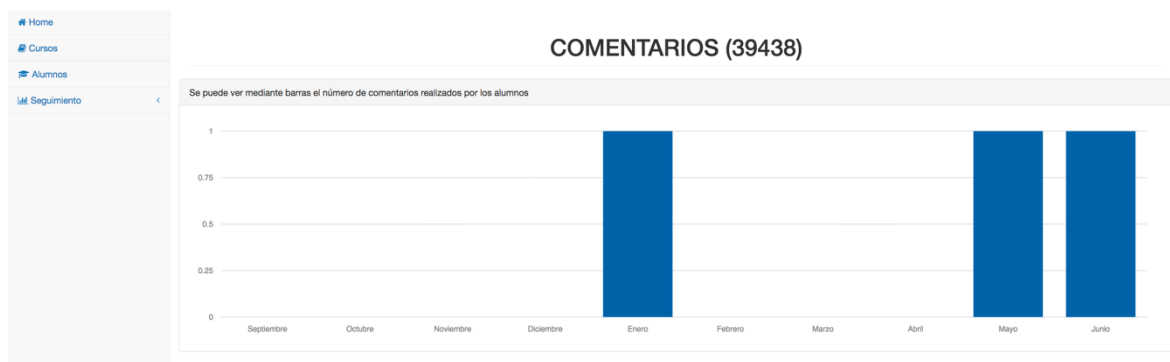


Figura 28: Comentarios individual

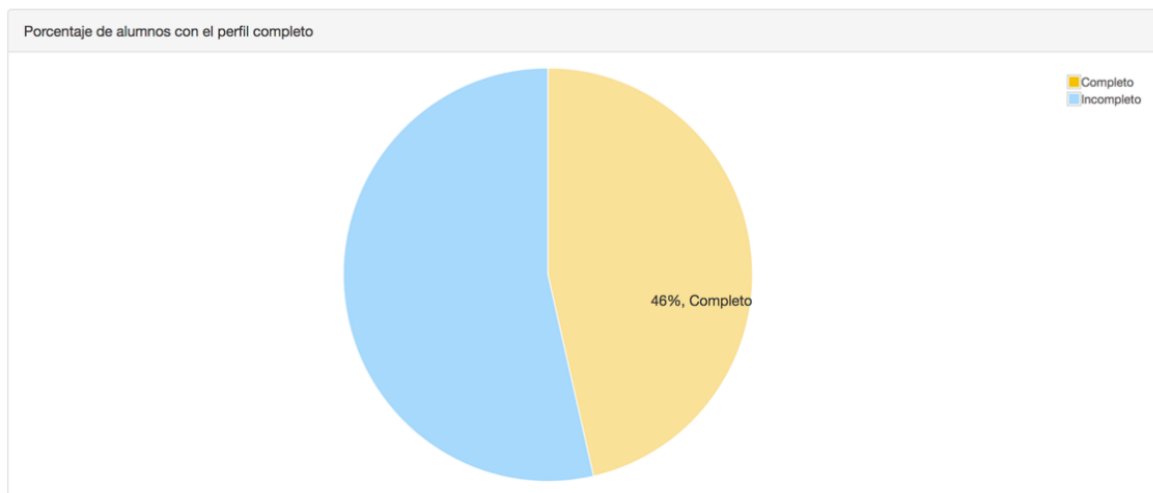


Figura 29: Perfiles general 1

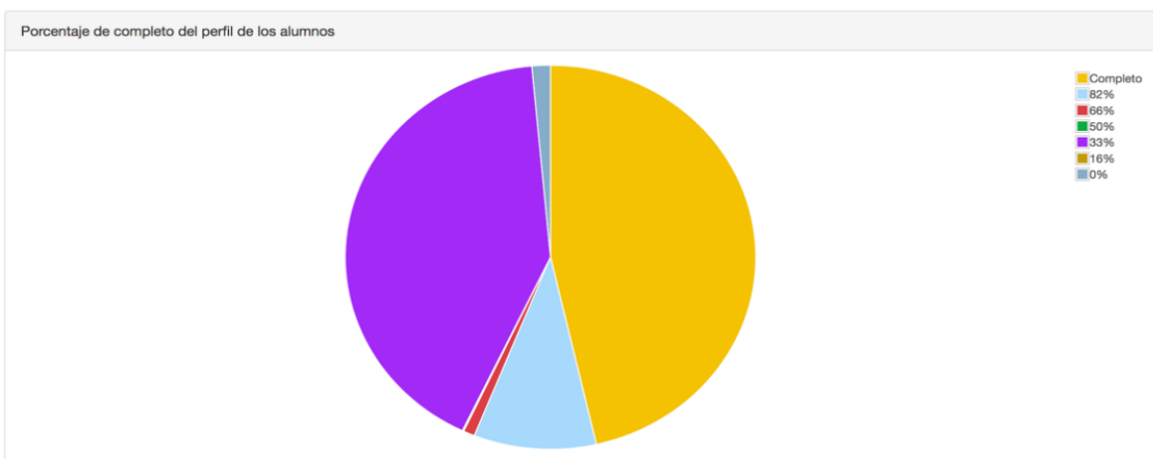


Figura 30: Perfiles general 2



Figura 31: Perfiles general 3

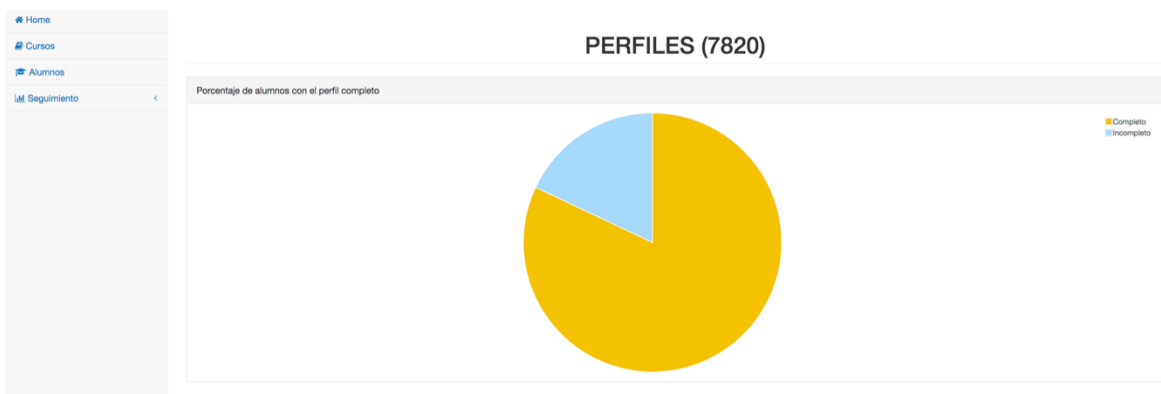


Figura 32: Perfiles individual

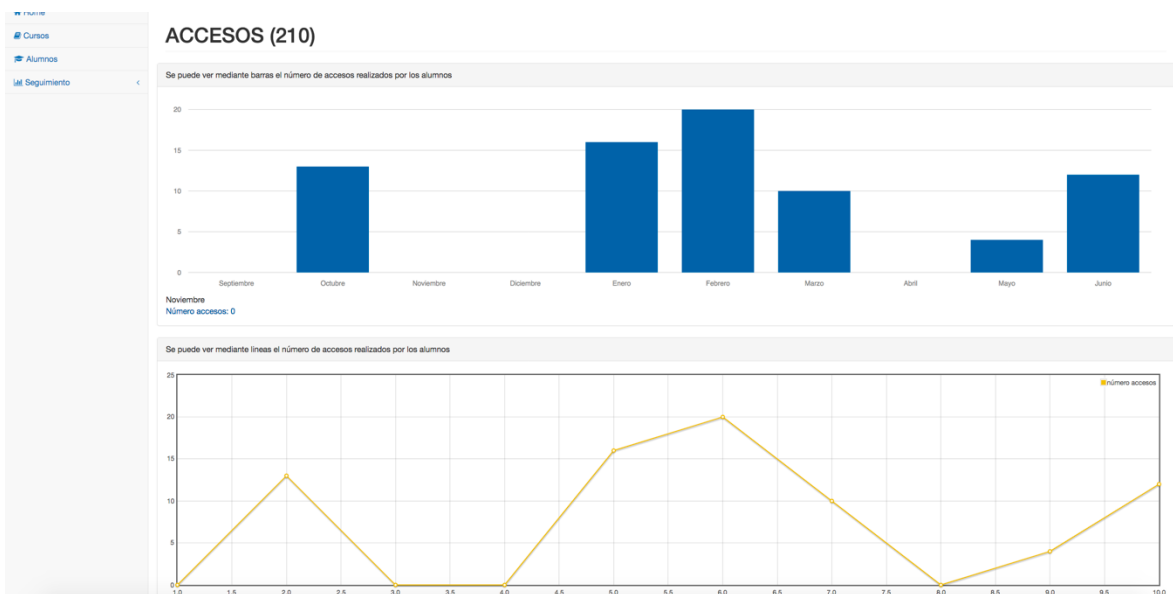


Figura 33: Accesos individual

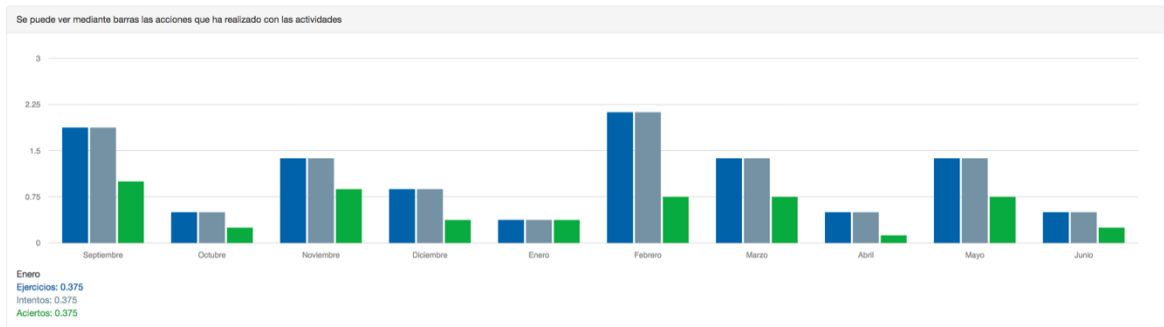


Figura 34: Actividades general 2

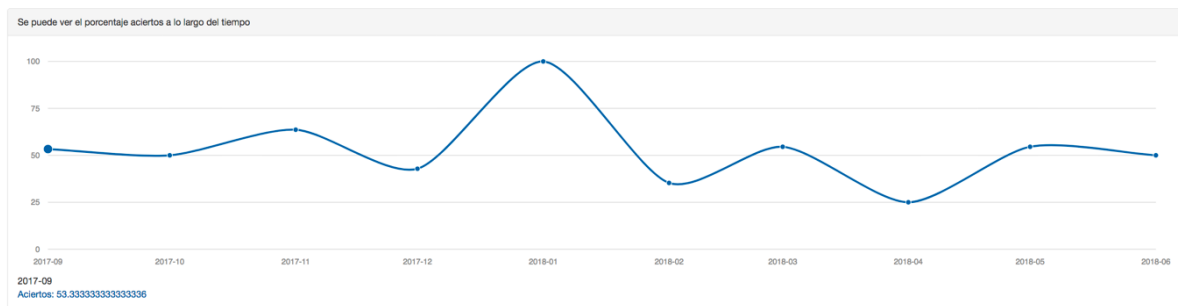


Figura 35: Actividades general 3

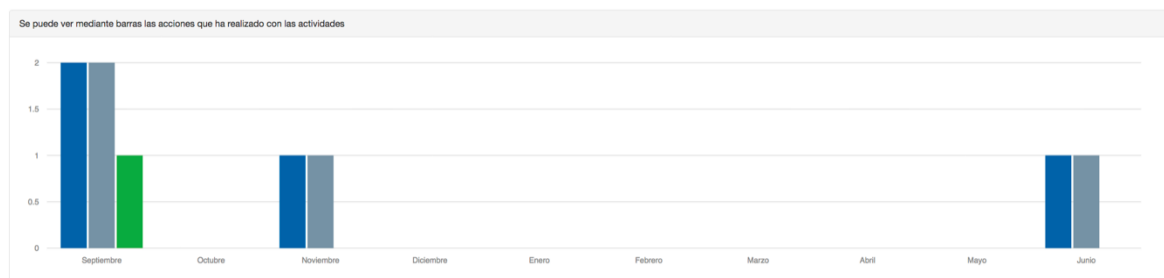


Figura 36: Actividades individuales 2

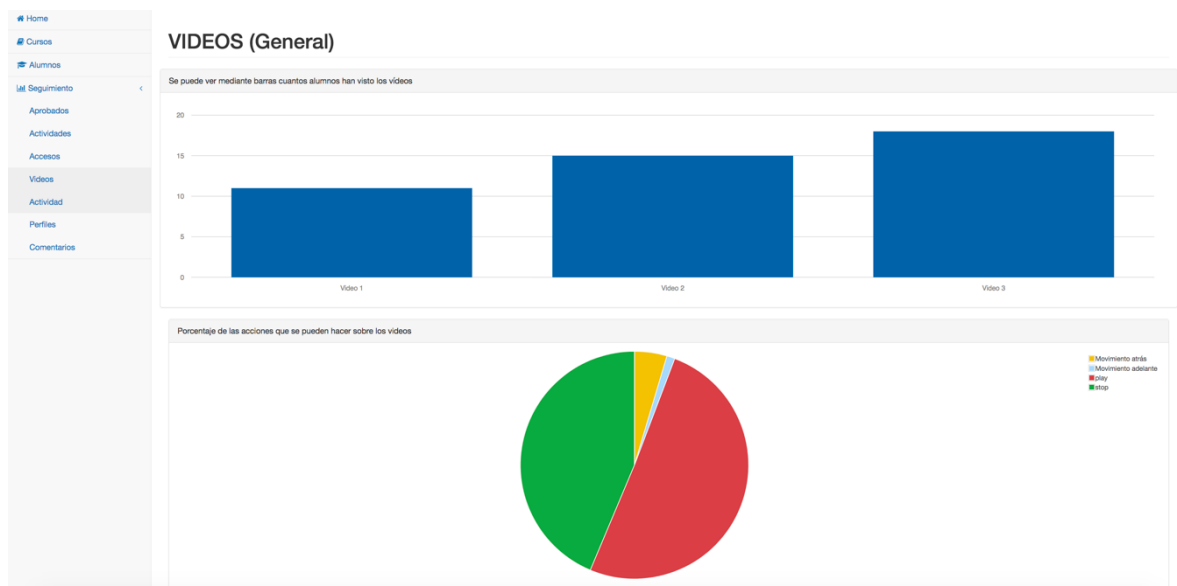


Figura 37: Videos general

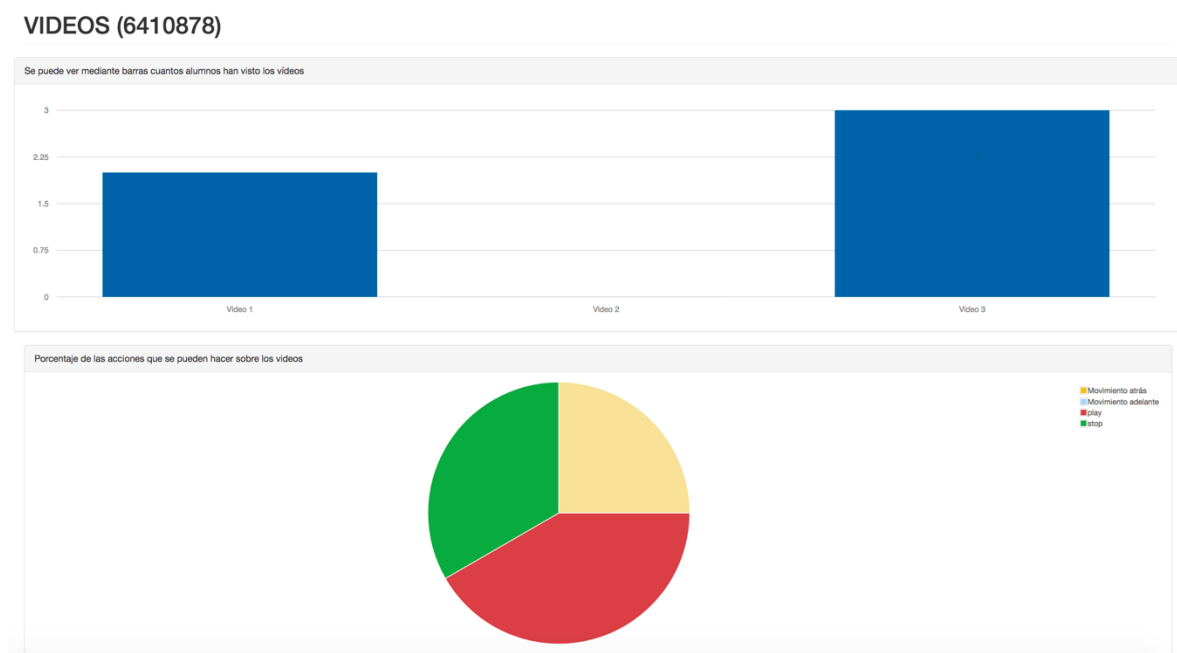


Figura 38: Videos individual

Actividades (General)

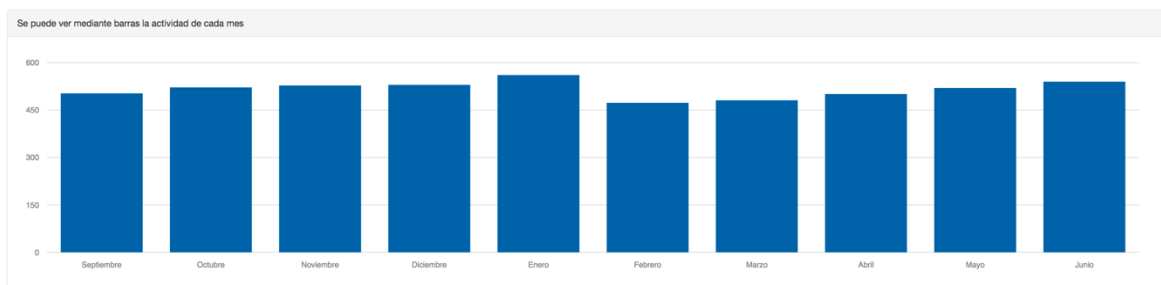


Figura 39: Actividad general

Actividades (6554784)

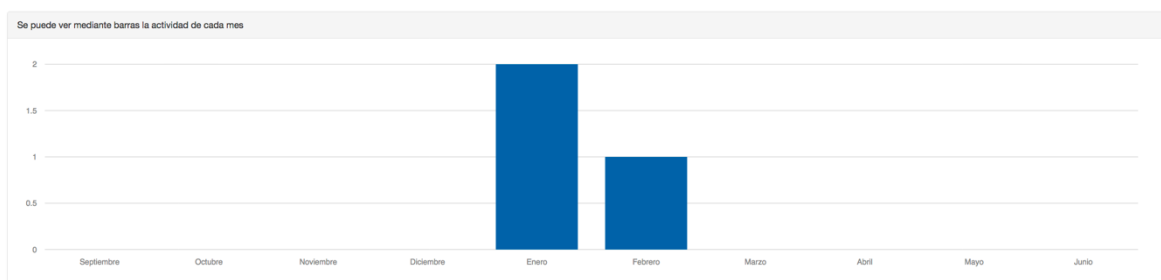


Figura 40: Actividad individual

